Materials Transactions, JIM, Vol. 40, No. 11 (1999), pp. 1186 to 1192
Special Issue on Advances in Computational Materials Science and Engineering

Acceleration of Self-Consistent Electronic-Structure Calculations: Storage-Saving and Multiple-Secant Implementation of the Broyden Method

Akitaka Sawamura*, Masanori Kohyama**, Tomohiro Keishi* and Mikio Kaji*

*Sumitomo Electric Industries, Ltd., Seika 619-0237, Japan
**Osaka National Research Institute, Ikeda 563-8577, Japan

On the basis of the Broyden method to solve simultaneous nonlinear equations, we present efficient computational schemes for acceleration of self-consistent electronic-structure calculations. The schemes, designed to achieve smoother and faster convergence on limited-storage conditions, involve a storage-saving representation of an approximate quasi-Newton matrix mathematically valid even when iteration history data are partially discarded. Moreover, to avoid numerical instability, a vector space where the multiple-secant condition is satisfied is determined dynamically. The efficiency and stability of the schemes is confirmed by the self-consistent electronic-structure calculations of a Si (011) surface model.

(Received May 19, 1999; In Final Form July 28, 1999)

Keywords: *silicon, electronic-structure calculation, pseudopotential, quasi-Newton method, Broyden method*

I. Introduction

In the past few years first-principles calculations based on the Kohn-Sham (KS) density functional theory[1] have gained enormous interest among solid-state physists, materials scientists, and quantum chemists. Successful applications of the KS theory to various properties of realistic materials and molecules can be found in the literatures[2]-[8]. This is due not only to reasonable accuracy of the local density approximation[9], but also to developments in algorithms to minimize the KS energy functional, or equivalently to obtain self-consistent solution of the KS equation. A common approach lying in the algorithms for the self-consistent solution, which can be embodied in terms of electron density[1], one-electron potential[10], or wavefunctions[11], is to define a computational scheme through which an approximate solution is reincarnated as a better one repeatedly until a given convergence criterion is satisfied.

The up-to-date self-consistency strategies, by which the KS energy functional is minimized with respect to the wavefunction, involving molecular-dymanics[11][12], steepest-descent[13][14], and conjugate-gradient[15]-[17] methods are the mathematically "cooler" and have won a great success in particular for large-scale applications. Kohyama[18] and Kresse and Furthmüller[19] have, however, pointed out that the traditional self-consistency strategies, by which the KS energy functional is minimized with respect to the electron density or the one-electron potential, are the more robust and versatile than those with respect to the wavefunction, because the former strategies are insensitive to fluctuation in electron occupancies at eigenstates near the Fermi level. This is true in particular when the strategies are equipped with preconditioned relaxation methods based on the Thomas-Fermi[20] or perturbation[21]-[23] theory, or with the Anderson method[24][25], the Broyden method[10][26][27] or its modified variant[28][29].

A cost which must be paid for the robustness and versatility of the traditional self-consistency strategies is that a gradient, a functional derivative of the KS energy with respect to the approximate solution, cannot be calculated easily and thus must be replaced by a difference between the input and output densities or potentials[10]. This leads to computational inconvenience that simple and efficient algorithms such as the conjugate-gradient method are not readily appliciable[30], because the Jacobian operator, through which the difference quantity is obtained from the input, is no more a Hermitian one[10][31]. Therefore as mentioned above, to accelerate convergence there have been adopted more complicated algorithms such as the Anderson and Broyden methods, which require iteration history data which consist of the input and difference quantities to be stored. When it takes many iterations to obtain the converged solution, a huge storage must be reserved for the iteration history data. This problem has motivated us to develop the Broyden-like computational scheme. Our scheme may resemble those proposed by Kresse and Furthmüller[19] and by Eyert[32], but is shown to be tolerant of the iteration history data partially discarded and in addition, incorporates the spirit of the Anderson method with a device against numerical instability to achieve smoother and faster convergence.

Our paper is organized as follows: In Sec. II we describe briefly the test problem which is used throughout the present study. In Sec. III we explain our storage-saving schemes based on the Broyden method

and examine them against the test problem. In Sec. IV we refine one of our storage-saving schemes by adding a multiple-secant feature and show the efficiency and stability of the resultant schemes. Finally in Sec. V we summarize the motivation, structure, and test results of our schemes and mention future implication.

II. Test Problem

A test problem for computational schemes in general should be easily implemented but also should faithfully represent physics of objects to which we intend to apply the schemes. We have decided to take the latter requirement seriously and chosen the self-consistent electronic-structure calculations of a Si (011) surface model as the test problem, because it is easy to implement the computational schemes which appear in the present study into an existing electronic-structure code.

First, we adopt the traditional self-consistency strategy where the self-consistent solution is sought in terms of the one-electron potential. Let us assume that we have an initial guess for the one-electron potential as the input quantity V_{in}. Then we solve the KS equation,

$$\left(-\frac{\hbar^2}{2m_e}\nabla^2+V_{\text{ext}}+V_{\text{in}}\right)\psi_n=\varepsilon_n\psi_n, \tag{1}$$

where $\hbar$ is the Planck constant, m_e is an electron mass, V_{ext} is an external potential representing constituent ions, ε_n is the nth lowest eigenvalue, and ψ_n is the corresponding wavefunction. We have utilized the Davidson-type algorithm[33] to solve eq. (1). The electron density ρ is obtained by,

$$\rho=2\sum_n f(\varepsilon_n-\varepsilon_f)|\psi_n|^2, \tag{2}$$

where ε_f is the Fermi energy and f is an appropriate smoothing function. From ρ the Hartree potential V_H is obtained by solving the Poisson equation,

$$\nabla^2 V_H=-8\pi\rho, \tag{3}$$

while an exchange-correlation potential V_{xc}, which represents quantum mechanical contribution to the electron interaction, is given by formulae roughly proportional to $\rho^{1/3}$ within the local density approximation. The output one-electron potential V_{out} is given by

$$V_{\text{out}}=V_H+V_{\text{xc}}. \tag{4}$$

The functional derivative of the KS energy E_{KS} with respect to the input potential is given by[10]

$$\frac{\delta E_{\text{KS}}}{\delta V_{\text{in}}}=\chi_0 V_{\text{diff}}, \tag{5}$$

where χ_0 is an independent-particle polarizability operator and V_{diff} is defined by,

$$V_{\text{diff}}=V_{\text{out}}-V_{\text{in}}. \tag{6}$$

The self-consistent solution V_{sc} may be defined as V_{in} which satisfies $\delta E_{\text{KS}}/\delta V_{\text{in}}=0$. Since it is computatinally demanding to evaluate χ_0 and thus $\delta E_{\text{KS}}/\delta V_{\text{in}}$, however, in practice V_{sc} is defined as V_{in} such that

$$V_{\text{diff}}=0 \tag{7}$$

is satisfied. Since eq. (7) is a nonlinear equation implicitly defined as a set of eqs. (1) to (6), it is solved by iterative schemes as explained in the next section.

The Si (011) surface model is a rectangular supercell containing five (011) and three empty layers, and thus ten Si atoms. The atoms are fixed at each ideal site. The size of the supercell is $5.43\times10^{-10}\,\text{m}\times3.84\times10^{-10}\,\text{m}\times1.536\times10^{-9}\,\text{m}$. The supercell is so small that the reader reexamine the results with ease but large enough to cause slow convergence with an inappropriate scheme as shown later. Wavefunctions are expanded by a plane-wave basis set within an energy cutoff of 2.58×10^{-17} J, while the electron density and potential of 1.03×10^{-16} J, which corresponds to 12051 plane waves. A soft norm-conserving pseudopotential[34][35] is adopted to avoid dealing explicitly with chemically inert core electrons.

Remaining aspects of the computational framework not described here are found in Refs.[3]-[7].

III. Storage-Saving Schemes Based on the Broyden Method

Considering discretization, we introduce a vector notation for the potentials, each of which is expanded by $\mathcal{N}$ Fourier components. Since as explained in the previous section the self-consistent calculation is reduced to solve eq. (7), if the Jacobian operator A, implicitly defined by

$$\vec{V}_{\text{diff}}=-A\vec{V}_{\text{in}}, \tag{8}$$

is available, the Newton-type iterative procedure,

$$\vec{V}_{\text{in}}^{(n+1)}=\vec{V}_{\text{in}}^{(n)}+[A^{(n)}]^{-1}\vec{V}_{\text{diff}}^{(n)}, \tag{9}$$

will lead to a vector sequence $\{\vec{V}_{\text{in}}^{(1)},\vec{V}_{\text{in}}^{(2)},\cdots,\vec{V}_{\text{in}}^{(n)}\}$ rapidly approaching $\vec{V}_{\text{sc}}$. Here superscripts in parentheses denote iteration numbers and $\vec{V}_{\text{diff}}^{(n)}$ and $A^{(n)}$ are the potential difference and the Jacobian operator calculated from $\vec{V}_{\text{in}}^{(n)}$, respectively. Due to convenience, we have changed the sign of A compared to common practice. Since explicit evaluation and handling of A are as demanding as of χ_0, however, Bendt and Zunger[10] have shown that when the inverse Jacobian $[A^{(n)}]^{-1}$ is recursively approximated by $B^{(n)}$ using Broyden's updating formula[26],

$$B^{(n)}=B^{(n-1)}-\frac{\{\delta\vec{V}_{\text{in}}^{(n-1)}+B^{(n-1)}\delta\vec{V}_{\text{diff}}^{(n-1)}\}\delta\vec{V}_{\text{diff}}^{(n-1)\dagger}}{\|\delta\vec{V}_{\text{diff}}^{(n-1)}\|^2}, \tag{10a}$$

with

$$\delta\vec{V}_{\text{in}}^{(n-1)}=\vec{V}_{\text{in}}^{(n)}-\vec{V}_{\text{in}}^{(n-1)},$$

and

$$\delta\vec{V}_{\text{diff}}^{(n-1)}=\vec{V}_{\text{diff}}^{(n)}-\vec{V}_{\text{diff}}^{(n-1)},$$

then the input potential for the next cycle predicted by

$$\vec{V}_{\text{in}}^{(n+1)}=\vec{V}_{\text{in}}^{(n)}+B^{(n)}\vec{V}_{\text{diff}}^{(n)}, \tag{10b}$$

converges toward the solution $\vec{V}_{\text{sc}}$ in fewer iterations than

that generated by a simple relaxation formula,

$$\vec{V}_{\rm in}^{(n+1)}=\vec{V}_{\rm in}^{(n)}+\beta\vec{V}_{\rm diff}^{(n)}, \tag{11}$$

where β is a mixing parameter. Note that by $\|\cdot\|$ we mean the Euclid norm of a vector. This efficiency originates from $\boldsymbol{B}^{(n)}$ so defined in eq. (10a) as to satisfy a secant condition,

$$\delta\vec{V}_{\rm in}^{(n-1)}=-\boldsymbol{B}^{(n)}\delta\vec{V}_{\rm diff}^{(n-1)}, \tag{12}$$

which is a mimic of an equation,

$$\delta\vec{V}_{\rm in}^{(n)}=-[\boldsymbol{A}^{(n)}]^{-1}\delta\vec{V}_{\rm diff}^{(n)}, \tag{13}$$

valid at the limit of $\|\delta\vec{V}_{\rm diff}^{(n)}\|\to 0$.

1. Description of storage-saving schemes

The Broyden method exemplified by eq. (10a) seemingly requires huge $\mathcal{N}\times\mathcal{N}$ matrices to be handled, but Srivastava[27] have developed an alternative computational scheme,

$$\vec{\xi}^{(n-1)}=\delta\vec{V}_{\rm in}^{(n-1)}+\left\{\boldsymbol{B}^{(1)}-\sum_{k=1}^{n-2}\frac{\vec{\xi}^{(k)}\delta\vec{V}_{\rm diff}^{(k)\dagger}}{\|\delta\vec{V}_{\rm diff}^{(k)}\|^2}\right\}\delta\vec{V}_{\rm diff}^{(n-1)}, \tag{14a}$$

and

$$\vec{V}_{\rm in}^{(n+1)}=\vec{V}_{\rm in}^{(n)}+\left\{\boldsymbol{B}^{(1)}-\sum_{k=1}^{n-1}\frac{\vec{\xi}^{(k)}\delta\vec{V}_{\rm diff}^{(k)\dagger}}{\|\delta\vec{V}_{\rm diff}^{(k)}\|^2}\right\}\vec{V}_{\rm diff}^{(n)}, \tag{14b}$$

where the approximate inverse Jacobian matrix $\boldsymbol{B}^{(n)}$ is implicitly expressed by a sum of an initial guess $\boldsymbol{B}^{(1)}$ and dyadic products of vectors. Srivastava's scheme represented by eqs. (14), where only vector sequences $\{\vec{V}_{\rm diff}^{(1)}, \vec{V}_{\rm diff}^{(2)},\cdots,\vec{V}_{\rm diff}^{(n-1)}\}$ and $\{\vec{\xi}^{(1)},\vec{\xi}^{(2)},\cdots,\vec{\xi}^{(n-1)}\}$ must be stored, is preferable if $\boldsymbol{B}^{(1)}$ for the inverse Jacobian matrix is easily dealt with and if the iteration count n is not too large.

Recently, Byrd, Nocedal, and Schnabel[36] (BNS) have proposed the other computational scheme where $\boldsymbol{B}^{(n)}$ is implicitly handled. In their scheme $\vec{V}_{\rm in}^{(n+1)}$ is given by

$$\vec{V}_{\rm in}^{(n+1)}=\vec{V}_{\rm in}^{(n)}+\boldsymbol{B}^{(1)}\{\boldsymbol{I}+\boldsymbol{Y}^{(n-1)}[\boldsymbol{Q}^{(n-1)}]^{-1}\boldsymbol{Y}^{(n-1)\dagger}\}\vec{V}_{\rm diff}^{(n)} +\boldsymbol{S}^{(n-1)}[\boldsymbol{Q}^{(n-1)}]^{-1}\boldsymbol{Y}^{(n-1)\dagger}\vec{V}_{\rm diff}^{(n)}, \tag{15a}$$

where $\boldsymbol{I}$ is an $\mathcal{N}\times\mathcal{N}$ identity matrix, $\boldsymbol{S}^{(n-1)}$ and $\boldsymbol{Y}^{(n-1)}$ are $\mathcal{N}\times(n-1)$ rectangular matrices defined by vector sequences,

$$\boldsymbol{S}^{(n-1)}=\{\delta\vec{V}_{\rm in}^{(1)},\delta\vec{V}_{\rm in}^{(2)},\cdots,\delta\vec{V}_{\rm in}^{(n-1)}\}, \tag{15b}$$

$$\boldsymbol{Y}^{(n-1)}=\{\delta\vec{V}_{\rm diff}^{(1)},\delta\vec{V}_{\rm diff}^{(2)},\cdots,\delta\vec{V}_{\rm diff}^{(n-1)}\}, \tag{15c}$$

respectively, and $\boldsymbol{Q}^{(n-1)}$ is a small $(n-1)\times(n-1)$ matrix

$$Q_{kl}^{(n-1)}=\begin{cases}\delta\vec{V}_{\rm diff}^{(k)\dagger}\delta\vec{V}_{\rm diff}^{(l)} & \text{if}\quad k\le l\\ 0 & \text{otherwise.}\end{cases} \tag{15d}$$

The rectangular matrices $\boldsymbol{S}^{(n-1)}$ and $\boldsymbol{Y}^{(n-1)}$ contain the iteration history data and thus must be stored. Along with Srivastava's, the BNS scheme is preferable if $\boldsymbol{B}^{(1)}$ is easily dealt with and if the iteration count n is not too large.

Equations (14) generate the vector sequence $\{\vec{V}_{\rm in}^{(1)},\vec{V}_{\rm in}^{(2)},\cdots,\vec{V}_{\rm in}^{(n)}\}$ which coincides with what is obtained by using eqs. (10) under exact arithmetic, as do eqs. (15). With this respect the two schemes, one is proposed by Srivastava and the other by BNS, are equivalent to each other. This is not the case, however, when the vector sequences for the iteration history data are no more fully kept in the computer storage.

Let only the history data of the v latest iterations be allowed to survive. When the iteration count n exceeds $v+1$, eqs. (14) exemplifying the Srivastava scheme can be modified into,

$$\vec{\xi}^{(n-1)}=\delta\vec{V}_{\rm in}^{(n-1)}+\left\{\boldsymbol{B}^{(1)}-\sum_{k=n-v}^{n-2}\frac{\vec{\xi}^{(k)}\delta\vec{V}_{\rm diff}^{(k)\dagger}}{\|\delta\vec{V}_{\rm diff}^{(k)}\|^2}\right\}\delta\vec{V}_{\rm diff}^{(n-1)}, \tag{16a}$$

and

$$\vec{V}_{\rm in}^{(n+1)}=\vec{V}_{\rm in}^{(n)}+\left\{\boldsymbol{B}^{(1)}-\sum_{k=n-v}^{n-1}\frac{\vec{\xi}^{(k)}\delta\vec{V}_{\rm diff}^{(k)\dagger}}{\|\delta\vec{V}_{\rm diff}^{(k)}\|^2}\right\}\vec{V}_{\rm diff}^{(n)}, \tag{16b}$$

with $\{\vec{V}_{\rm diff}^{(n-v)},\vec{V}_{\rm diff}^{(n-v+1)},\cdots,\vec{V}_{\rm diff}^{(n-1)}\}$ and $\{\vec{\xi}^{(n-v)},\vec{\xi}^{(n-v+1)},\cdots,\vec{\xi}^{(n-1)}\}$ stored. Unfortunately, however, the scheme based on eqs. (16) deviates from the original Broyden method, because $\vec{\xi}^{(k)}$ appearing in the right-hand sides of eq. (16a) and (16b) is implicitly dependent on $\vec{\xi}^{(l)}$ with $l<n-v$ through earlier iteration cycles. This inexactitude will result in inefficiency as shown later. The limited-storage analogue of eqs. (15) are given by

$$\vec{V}_{\rm in}^{(n+1)}=\vec{V}_{\rm in}^{(n)}+\boldsymbol{B}^{(1)}\{\boldsymbol{I}+\boldsymbol{Y}_{(n-v)}^{(n-1)}[\boldsymbol{Q}_{(n-v)}^{(n-1)}]^{-1}\boldsymbol{Y}_{(n-v)}^{(n-1)\dagger}\}\vec{V}_{\rm diff}^{(n)} +\boldsymbol{S}_{(n-v)}^{(n-1)}[\boldsymbol{Q}_{(n-v)}^{(n-1)}]^{-1}\boldsymbol{Y}_{(n-v)}^{(n-1)\dagger}\vec{V}_{\rm diff}^{(n)}, \tag{17a}$$

where $\boldsymbol{S}_{(n-v)}^{(n-1)}$ and $\boldsymbol{Y}_{(n-v)}^{(n-1)}$ are $\mathcal{N}\times v$ rectangular matrices defined by vector sequences,

$$\boldsymbol{S}_{(n-v)}^{(n-1)}=\{\delta\vec{V}_{\rm in}^{(n-v)},\delta\vec{V}_{\rm in}^{(n-v+1)},\cdots,\delta\vec{V}_{\rm in}^{(n-1)}\}, \tag{17b}$$

$$\boldsymbol{Y}_{(n-v)}^{(n-1)}=\{\delta\vec{V}_{\rm diff}^{(n-v)},\delta\vec{V}_{\rm diff}^{(n-v+1)},\cdots,\delta\vec{V}_{\rm diff}^{(n-1)}\}, \tag{17c}$$

respectively, and $\boldsymbol{Q}_{(n-v)}^{(n-1)}$ is a $v\times v$ matrix

$$Q_{(n-v)kl}^{(n-1)}=\begin{cases}\delta\vec{V}_{\rm diff}^{(k-n+v+1)\dagger}\delta\vec{V}_{\rm diff}^{(l-n+v+1)} & \text{if}\quad k\le l\\ 0 & \text{otherwise.}\end{cases} \tag{17d}$$

In contrast with eqs. (16), eqs. (17) lead to a storage-saving scheme which remains exactly the Broyden method where the approximate inverse Jacobian matrix is updated using information of the v latest iterations only. This is shown as follows: Let us substitute $p+v+1$ for n in eqs. (17). Then we have

$$\vec{V}_{\rm in}^{(p+v+2)}=\vec{V}_{\rm in}^{(p+v+1)}+\boldsymbol{B}^{(1)}\{\boldsymbol{I}+\boldsymbol{Y}_{(p+1)}^{(p+v)}[\boldsymbol{Q}_{(p+1)}^{(p+v)}]^{-1}\boldsymbol{Y}_{(p+1)}^{(p+v)\dagger}\} \times\vec{V}_{\rm diff}^{(p+v+1)}+\boldsymbol{S}_{(p+1)}^{(p+v)}[\boldsymbol{Q}_{(p+1)}^{(p+v)}]^{-1}\boldsymbol{Y}_{(p+1)}^{(p+v)\dagger}\vec{V}_{\rm diff}^{(p+v+1)}, \tag{18a}$$

with

$$\boldsymbol{S}_{(p+1)}^{(p+v)}=\{\delta\vec{V}_{\rm in}^{(p+1)},\delta\vec{V}_{\rm in}^{(p+2)},\cdots,\delta\vec{V}_{\rm in}^{(p+v)}\}, \tag{18b}$$

$$\boldsymbol{Y}_{(p+1)}^{(p+v)}=\{\delta\vec{V}_{\rm diff}^{(p+1)},\delta\vec{V}_{\rm diff}^{(p+2)},\cdots,\delta\vec{V}_{\rm diff}^{(p+v)}\}, \tag{18c}$$

and

$$Q_{(p+1)kl}^{(p+v)}=\begin{cases}\delta\vec{V}_{\rm diff}^{(k-p)\dagger}\delta\vec{V}_{\rm diff}^{(l-p)} & \text{if}\quad k\le l\\ 0 & \text{otherwise.}\end{cases} \tag{18d}$$

Clearly, if in eqs. (18) we substitute 0 for p and $n-1$ for ν, which is nothing but a shift in definition of the iteration index, then eqs. (18) become formally equivalent to eqs. (15). Therefore, the scheme represented by eqs. (17) is the exact incarnation of Broyden's idea despite its storage-saving feature.

2. Comparison of storage-saving schemes

Before turning to the comparison of the storage-saving schemes immediately, we show the equations which are actually used in our computer program. The storage-saving Srivastava (SSS) scheme is given by a set of equations,

$$\vec{\xi}^{(n-1)}=\delta\vec{V}_{\rm in}^{(n-1)}+\left\{\boldsymbol{B}^{(1)}-\sum_{k=n-\tilde{\nu}}^{n-2}\vec{\xi}^{(k)}\vec{\mathcal{V}}^{(k)\dagger}\right\}\delta\vec{V}_{\rm diff}^{(n-1)}, \tag{19a}$$

and

$$\vec{V}_{\rm in}^{(n+1)}=\vec{V}_{\rm in}^{(n)}+\left\{\boldsymbol{B}^{(1)}-\sum_{k=n-\tilde{\nu}}^{n-1}\vec{\xi}^{(k)}\vec{\mathcal{V}}^{(k)\dagger}\right\}\vec{V}_{\rm diff}^{(n)}, \tag{19b}$$

where $\vec{\mathcal{V}}^{(k)}$ defined by

$$\vec{\mathcal{V}}^{(k)}=\frac{\delta\vec{V}_{\rm diff}^{(k)}}{\|\delta\vec{V}_{\rm diff}^{(k)}\|^2}, \tag{19c}$$

is introduced to avoid unnecessary evaluation of $\|\delta\vec{V}_{\rm diff}^{(k)}\|^2$. The vector sequences $\{\vec{\mathcal{V}}^{(n-\tilde{\nu})}, \vec{\mathcal{V}}^{(n-\tilde{\nu}+1)}, \cdots, \vec{\mathcal{V}}^{(n-1)}\}$ and $\{\vec{\xi}^{(n-\tilde{\nu})}, \vec{\xi}^{(n-\tilde{\nu}+1)}, \cdots, \vec{\xi}^{(n-1)}\}$ must be stored and kept unchanged outside subroutines embodying eqs. (19). $\tilde{\nu}$ is given by,

$$\tilde{\nu}=\begin{cases} n-1 & \text{if} \quad n\le\nu+1 \\ \nu & \text{otherwise,}\end{cases} \tag{20}$$

where ν is the history data limit as introduced previously. The core equation of the storage-saving BNS (SSBNS) scheme is,

$$\begin{aligned}\vec{V}_{\rm in}^{(n+1)}=\vec{V}_{\rm in}^{(n)}+\boldsymbol{B}^{(1)}\{\boldsymbol{I}+\boldsymbol{Y}_{(n-\tilde{\nu})}^{(n-1)}[\boldsymbol{Q}_{(n-\tilde{\nu})}^{(n-1)}]^{-1}\boldsymbol{Y}_{(n-\tilde{\nu})}^{(n-1)\dagger}\}\\ \times\vec{V}_{\rm diff}^{(n)}+\boldsymbol{S}_{(n-\tilde{\nu})}^{(n-1)}[\boldsymbol{Q}_{(n-\tilde{\nu})}^{(n-1)}]^{-1}\boldsymbol{Y}_{(n-\tilde{\nu})}^{(n-1)\dagger}\vec{V}_{\rm diff}^{(n)},\end{aligned} \tag{21a}$$

where $\boldsymbol{S}_{(n-\tilde{\nu})}^{(n-1)}$ and $\boldsymbol{Y}_{(n-\tilde{\nu})}^{(n-1)}$ are $\mathcal{N}\times\tilde{\nu}$ rectangular matrices,

$$\boldsymbol{S}_{(n-\tilde{\nu})}^{(n-1)}=\left\{\frac{\delta\vec{V}_{\rm in}^{(n-\tilde{\nu})}}{\|\delta\vec{V}_{\rm diff}^{(n-\tilde{\nu})}\|}, \frac{\delta\vec{V}_{\rm in}^{(n-\tilde{\nu}+1)}}{\|\delta\vec{V}_{\rm diff}^{(n-\tilde{\nu}+1)}\|}, \cdots, \frac{\delta\vec{V}_{\rm in}^{(n-1)}}{\|\delta\vec{V}_{\rm diff}^{(n-1)}\|}\right\}, \tag{21b}$$

and

$$\boldsymbol{Y}_{(n-\tilde{\nu})}^{(n-1)}=\left\{\frac{\delta\vec{V}_{\rm diff}^{(n-\tilde{\nu})}}{\|\delta\vec{V}_{\rm diff}^{(n-\tilde{\nu})}\|}, \frac{\delta\vec{V}_{\rm diff}^{(n-\tilde{\nu}+1)}}{\|\delta\vec{V}_{\rm diff}^{(n-\tilde{\nu}+1)}\|}, \cdots, \frac{\delta\vec{V}_{\rm diff}^{(n-1)}}{\|\delta\vec{V}_{\rm diff}^{(n-1)}\|}\right\}, \tag{21c}$$

respectively, and $\boldsymbol{Q}_{(n-\tilde{\nu})}^{(n-1)}$ is a $\tilde{\nu}\times\tilde{\nu}$ matrix

$$Q_{(n-\tilde{\nu})kl}^{(n-1)}=\begin{cases}\dfrac{\delta\vec{V}_{\rm diff}^{(k-n+\tilde{\nu}+1)\dagger}\delta\vec{V}_{\rm diff}^{(l-n+\tilde{\nu}+1)}}{\|\delta\vec{V}_{\rm diff}^{(k-n+\tilde{\nu}+1)}\|\|\delta\vec{V}_{\rm diff}^{(l-n+\tilde{\nu}+1)}\|} & \text{if} \quad k\le l \\ 0 & \text{otherwise.}\end{cases} \tag{21d}$$

$\tilde{\nu}$ is given by eq. (20) as with the SSS scheme. Note that in the actual program the matrices $\boldsymbol{S}_{(n-\tilde{\nu})}^{(n-1)}$, $\boldsymbol{Y}_{(n-\tilde{\nu})}^{(n-1)}$, and $\boldsymbol{Q}_{(n-\tilde{\nu})}^{(n-1)}$ are defined as normalized quantities to enhance numerical stability.

Table 1 The required iteration counts to reach self-consistency using the storage-saving Srivastava-type and storage-saving Byrd-Nocedal-Schnabel schemes with various history data limits.

Scheme	History data limit					
	3	6	9	12	15	∞
storage-saving Srivastava	35	26	27	31	23	22
storage-saving Byrd-Nocedal-Schnabel	25	24	24	22	22	22

The test problem is already described in the previous section, but here three features in it are furthermore specified. First, the input potential at the first iteration, $\vec{V}_{\rm in}^{(1)}$, is calculated from a superposition of valence electron density of neutral silicon atoms. Second, the initial guess for the inverse Jacobian is give by,

$$\boldsymbol{B}^{(1)}=0.7\boldsymbol{I} \tag{22}$$

and allowed to be scaled only at the second iteration so that

$$\|\delta\vec{V}_{\rm in}^{(1)}+\boldsymbol{B}^{(1)}\delta\vec{V}_{\rm diff}^{(1)}\| \tag{23}$$

is minimized. Third, the self-consistency is regarded as achieved when all the Fourier components of the potential difference $\delta\vec{V}_{\rm diff}^{(n)}$ fall within 2.2×10^{-26} J, which is a tight criterion.

As mentioned above, we have expected that when a small amount of the computer storage is reserved for the iteration history data, the SSBNS scheme has an advantage of faster convergence over the SSS. This is confirmed by the actual calculations. **Table 1** shows how many iterations are performed to obtain the self-consistent solution with various history data limits. Clearly, unless the history data limit is removed, the SSBNS scheme leads to the faster convergence consistently. On the other hand, as misgiven, the SSS scheme is inferior not only in the convergence, but also in stability, because increasing the limit does not always result in smaller iteration counts. Therefore using the SSBNS scheme is recommendable.

IV. Multiple-Secant Schemes Based on the Broyden Method

1. Description of multiple-secant schemes

In the present section we propose further improved schemes. As already mentioned, the secant condition given by eq. (12) is responsible for the fast convergence achieved by using the schemes based on the Broyden method. Unfortunately, however, when the approximate inverse Jacobian is updated, an analogous equation,

$$\delta\vec{V}_{\rm in}^{(n-1)}=-\boldsymbol{B}^{(n+1)}\delta\vec{V}_{\rm diff}^{(n-1)}, \tag{24}$$

is no more satisfied. Therefore, it is quite natural to expect that a scheme where a multiple-secant condition,

$$\delta\vec{V}_{\rm in}^{(k)}=-\boldsymbol{B}^{(n)}\delta\vec{V}_{\rm diff}^{(k)}, \tag{25}$$

with $k\le n-1$ holds exactly[37]-[39] or approximately[28][29]

will lead to even faster convergence.

In the previous section we have shown that when the iteration history data are not kept fully, the SSBNS scheme has an advantage of the faster and more stable convergence. Another advantage that the multiple-secant condition can be straightfowardly incorporated into the BNS scheme[36], and thus into the SSBNS scheme. The multiple-secant variant of the SSBNS scheme, referred to as MSBNS scheme hereafter, is given by eqs. (21) except for the $v \times v$ matrix $\boldsymbol{Q}_{(n-\tilde{v})}^{(n-1)}$, which is defined by

$$\boldsymbol{Q}_{(n-\tilde{v})}^{(n-1)} = \boldsymbol{Y}_{(n-\tilde{v})}^{(n-1)\dagger} \boldsymbol{Y}_{(n-\tilde{v})}^{(n-1)}. \tag{26}$$

The MSBNS scheme satisfies the multiple-secant condition exactly. We will concentrate on the MSBNS and related scheme, because though the scheme with the multiple-secant condition satisfied approximately has been proposed and utilized in the electronic-structure calculations[29], it is less efficient than that where the multiple-secant condition holds exactly[32].

We show that the multiple-secant variant of the Broyden method can be regarded as Anderson's[24][25] nonlinear least-square method. This has been already pointed out[19][32], but it is important to remind us the equivalence. Suppose that we have already obtained the vector sequences, $\boldsymbol{S}_{(n-\tilde{v})}^{(n-1)}$ and $\boldsymbol{Y}_{(n-\tilde{v})}^{(n-1)}$ defined by eqs. (21b) and (21c), respectively. Then the input quantity $\vec{V}_{\text{in}}^{(n+1)}$ is predicted by the Anderson method as follows: within the subspace spanned by the vectors $\{\delta\vec{V}_{\text{in}}^{(n-\tilde{v})}/\|\delta\vec{V}_{\text{diff}}^{(n-\tilde{v})}\|, \delta\vec{V}_{\text{in}}^{(n-\tilde{v}+1)}/\|\delta\vec{V}_{\text{diff}}^{(n-\tilde{v}+1)}\|, \cdots, \delta\vec{V}_{\text{in}}^{(n-1)}/\|\delta\vec{V}_{\text{diff}}^{(n-1)}\|, \vec{V}_{\text{in}}^{(n)}\}$, the best approximate solution $\vec{V}_{\text{in}}^{(n)'}$ is constructed as,

$$\vec{V}_{\text{in}}^{(n)'} = \vec{V}_{\text{in}}^{(n)} + \sum_{k=n-\tilde{v}}^{n-1} \eta_k^{(n)} \frac{\delta\vec{V}_{\text{in}}^{(k)}}{\|\delta\vec{V}_{\text{diff}}^{(k)}\|}, \tag{27}$$

by requiring that a predicted difference vector

$$\vec{V}_{\text{diff}}^{(n)'} = \vec{V}_{\text{diff}}^{(n)} + \sum_{k=n-\tilde{v}}^{n-1} \eta_k^{(n)} \frac{\delta\vec{V}_{\text{diff}}^{(k)}}{\|\delta\vec{V}_{\text{diff}}^{(k)}\|}, \tag{28}$$

approximates a zero vector in a least-square sense. The predicted vector $\vec{V}_{\text{in}}^{(n)'}$ might be an adequate candidate for the next input quantity $\vec{V}_{\text{in}}^{(n+1)}$. In reality, however, this is not the case, because the vector $\vec{V}_{\text{in}}^{(n)'}$ does not expand the input vector space from which $\vec{V}_{\text{in}}^{(n)'}$ is extracted. Therefore, the next input quantity $\vec{V}_{\text{in}}^{(n+1)}$ should be given by

$$\vec{V}_{\text{in}}^{(n+1)} = \vec{V}_{\text{in}}^{(n)'} + \boldsymbol{B}^{(1)} \vec{V}_{\text{diff}}^{(n)'}, \tag{29}$$

where $\boldsymbol{B}^{(1)}$ is an approximation to the inverse Jacobian as appeared in the previous section. The coefficients $\eta_k^{(n)}$ are determined so that $\|\vec{V}_{\text{diff}}^{(n)'}\|$ is minimized. This leads to a linear least-square problem equivalent to solving a simultaneous linear equation[40],

$$\begin{pmatrix} \delta v_{n-\tilde{v},n-\tilde{v}} & \delta v_{n-\tilde{v},n-\tilde{v}+1} & \cdots & \delta v_{n-\tilde{v},n-1} \\ \delta v_{n-\tilde{v}+1,n-\tilde{v}} & \delta v_{n-\tilde{v}+1,n-\tilde{v}+1} & \cdots & \delta v_{n-\tilde{v}+1,n-1} \\ \vdots & \vdots & & \vdots \\ \delta v_{n-1,n-\tilde{v}} & \delta v_{n-1,n-\tilde{v}+1} & \cdots & \delta v_{n-1,n-1} \end{pmatrix} \begin{pmatrix} \eta_{n-\tilde{v}}^{(n)} \\ \eta_{n-\tilde{v}+1}^{(n)} \\ \vdots \\ \eta_{n-1}^{(n)} \end{pmatrix} = \begin{pmatrix} u_{n-\tilde{v}}^{(n)} \\ u_{n-\tilde{v}+1}^{(n)} \\ \vdots \\ \eta_{n-1}^{(n)} \end{pmatrix}, \tag{30a}$$

with

$$\delta v_{k,l} = \delta\vec{V}_{\text{diff}}^{(k)\dagger} \delta\vec{V}_{\text{diff}}^{(l)}, \tag{30b}$$

and

$$u_k^{(n)} = \delta\vec{V}_{\text{diff}}^{(k)\dagger} \vec{V}_{\text{diff}}^{(n)}. \tag{30c}$$

If to write down $\vec{V}_{\text{in}}^{(n+1)}$ explicitly we combine eqs. (27) to (30), we have

$$\vec{V}_{\text{in}}^{(n+1)} = \vec{V}_{\text{in}}^{(n)} + \boldsymbol{B}^{(1)} \vec{V}_{\text{diff}}^{(n)} + \{\boldsymbol{S}_{(n-\tilde{v})}^{(n-1)} + \boldsymbol{B}^{(1)} \boldsymbol{Y}_{(n-\tilde{v})}^{(n-1)}\} \times [\boldsymbol{Q}_{(n-\tilde{v})}^{(n-1)}]^{-1} \boldsymbol{Y}_{(n-\tilde{v})}^{(n-1)\dagger} \vec{V}_{\text{diff}}^{(n)}, \tag{31}$$

where $\boldsymbol{S}_{(n-\tilde{v})}^{(n-1)}$, $\boldsymbol{Y}_{(n-\tilde{v})}^{(n-1)}$, and $\boldsymbol{Q}_{(n-\tilde{v})}^{(n-1)}$ are exactly the same as those defined by eqs. (21b), (21c), and (26), respectively. To derive eq. (31) we have used the fact that the matrix appearing in the left-hand side of eq. (30a) is the same as $\boldsymbol{Q}_{(n-\tilde{v})}^{(n-1)}$ given by eq. (26). Obviously, eq. (31) is equivalent to eq. (21d) and thus the multiple-secant variant of the Broyden method can be regarded as the Anderson method.

It is still under debate whether the Anderson method, or equivalently the multiple-secant variant of the Broyden method is superior to the original Broyden method in the actual calculations[19][29]. From the purely mathematical viewpoint the former would be better than the latter. Thus if this is not the case, numerical instability is the probable cause. Since inversion of the matrix $\boldsymbol{Q}_{(n-\tilde{v})}^{(n-1)}$ given by eq. (21d) is well-defined for any $\boldsymbol{Y}_{(n-\tilde{v})}^{(n-1)}$ but that by eq. (26) is not always so, the multiple-secant variant of the Broyden method is vulnerable to linear dependence among the vectors $\{\delta\vec{V}_{\text{diff}}^{(n-\tilde{v})}/\|\delta\vec{V}_{\text{diff}}^{(n-\tilde{v})}\|, \delta\vec{V}_{\text{diff}}^{(n-\tilde{v}+1)}/\|\delta\vec{V}_{\text{diff}}^{(n-\tilde{v}+1)}\|, \cdots, \delta\vec{V}_{\text{diff}}^{(n-1)}/\|\delta\vec{V}_{\text{diff}}^{(n-1)}\|\}$. Setting the limit that the information of the latest $\tilde{v}$ iterations are allowed to be stored as in the previous section may cure the multiple-secant method of the numerical instability. The limit has to be chosen with great care, however. When the limit is too loose, the numerical instability cannot be avoided. On the contrary if the limit is too tight, the MSBNS scheme cannot demonstrate its potential fully.

Instead of setting the limit $\tilde{v}$ by hand to avoid the numerical instability rather than to economize the computer storage, we introduce another limit μ which we decide automatically by estimating the degree of linear dependence among $\{\delta\vec{V}_{\text{diff}}^{(n-\tilde{v})}/\|\delta\vec{V}_{\text{diff}}^{(n-\tilde{v})}\|, \delta\vec{V}_{\text{diff}}^{(n-\tilde{v}+1)}/\|\delta\vec{V}_{\text{diff}}^{(n-\tilde{v}+1)}\|, \cdots, \delta\vec{V}_{\text{diff}}^{(n-1)}/\|\delta\vec{V}_{\text{diff}}^{(n-1)}\|\}$ and then modify the difinition of $\boldsymbol{Q}_{(n-\tilde{v})}^{(n-1)}$. The degree of linear dependence is measured numerically by a condition number[41] of $\boldsymbol{Q}_{(n-\tilde{v})}^{(n-1)}$ defined by eq. (26). In general, the condition number $\kappa(\boldsymbol{A})$ with a matrix $\boldsymbol{A}$ is given by

$$\kappa(A)=\|A\|\|A^{-1}\|. \tag{32}$$

Here the norm for a matrix A is given by

$$\|A\|=\max_{\|x\|=1}\|Ax\|, \tag{33}$$

where x is any vector and $\|x\|$ and $\|Ax\|$ are vector norms as already appeared. If $\kappa(Q^{(n-1)}_{(n-\tilde{\nu})})$ is smaller than a specified value $\tilde{\kappa}$, the input vector $\vec{V}^{(n+1)}_{\text{in}}$ is calculated by eqs. (21) except for the $\tilde{\nu}\times\tilde{\nu}$ matrix $Q^{(n-1)}_{(n-\tilde{\nu})}$ defined by eq. (26). Otherwise we introduce new definitions of $Q^{(n-1)}_{(n-\tilde{\nu})}$ covering both eqs. (21d) and (26) on extreme occasions. For the first definition, $Q^{(n-1)}_{(n-\tilde{\nu})}$ is given by

$$Q^{(n-1)}_{(n-\tilde{\nu})kl}=\begin{cases}\dfrac{\delta\vec{V}^{(k-n+\tilde{\nu}+1)\dagger}_{\text{diff}}\delta\vec{V}^{(l-n+\tilde{\nu}+1)}_{\text{diff}}}{\|\delta\vec{V}^{(k-n+\tilde{\nu}+1)}_{\text{diff}}\|\,\|\delta\vec{V}^{(l-n+\tilde{\nu}+1)}_{\text{diff}}\|} & \\ \qquad\text{if}\quad k\le l\quad\text{or}\quad \mu\le l & \\ 0\qquad\text{otherwise,} & \end{cases} \tag{34}$$

where μ is the minimum positive integer so determined that $\kappa(Q^{(n-1)}_{(n-\tilde{\nu})})$ satisfies

$$\kappa(Q^{(n-1)}_{(n-\tilde{\nu})})<\tilde{\kappa}. \tag{35}$$

Here $\tilde{\kappa}$ is a maximum condition number which we have to prescribe. By combining eq. (34) with eqs. (21a), (21b), and (21c) we have a set of formulae where the latest $\tilde{\nu}-\mu+1$ iterations and the older remainder influence the determination of $\vec{V}^{(n+1)}_{\text{in}}$ through the multiple-secant variant and the original formulation, respectively, of the Broyden method. For the second definition, $Q^{(n-1)}_{(n-\tilde{\nu})}$ is given by

$$Q^{(n-1)}_{(n-\tilde{\nu})kl}=\begin{cases}\dfrac{\delta\vec{V}^{(k-n+\tilde{\nu}+1)\dagger}_{\text{diff}}\delta\vec{V}^{(l-n+\tilde{\nu}+1)}_{\text{diff}}}{\|\delta\vec{V}^{(k-n+\tilde{\nu}+1)}_{\text{diff}}\|\,\|\delta\vec{V}^{(l-n+\tilde{\nu}+1)}_{\text{diff}}\|} & \\ \qquad\text{if}\quad \mu\le k\quad\text{and}\quad \mu\le l & \\ \infty\qquad\text{if}\quad \mu>k=l & \\ 0\qquad\text{otherwise,} & \end{cases} \tag{36}$$

with μ determined as above. Equation (36) leads to prediciting $\vec{V}^{(n+1)}_{\text{in}}$ by the multiple-secant Broyden method referring only to the information retrieved from the latest $\tilde{\nu}-\mu+1$ iterations. For both cases, when eq. (35) is violated with any μ, we define $Q^{(n-1)}_{(n-\tilde{\nu})}$ by eq. (21d), that is, return to the original Broyden method.

2. Comparison of multiple-secant schemes

We have performed the same electronic structure calculations as in the previous section except that we have used the multiple-secant schemes, called MSBNS(1) and MSBNS(2) hereafter, to which eqs. (34) and (36) lead, respectively. $\tilde{\kappa}$ in eq. (35) is set to be 10^5. As already mentioned, these schemes are obtained by modifying the definition of $Q^{(n-1)}_{(n-\tilde{\nu})}$ in eqs. (21) for efficiency without sacrificing robustness. We expect these schemes to yield faster convergence toward the self-consistent solution than the SSS and SSBNS, which are examined in the previous section. In addition, if $\tilde{\kappa}$ is not too small, we expect the MSBNS(1) and MSBNS(2) schemes to exhibit similar convergence properties, because while overridden in the MSBNS(1), information contained in the $(n-\tilde{\nu})$th iteration to the $(n-\tilde{\nu}+\mu-1)$th is simply ignored in the MSBNS(2). It is left to actual calculations which scheme performs better.

Table 2 The required iteration counts to reach self-consistency using the multiple-secant Byrd-Nocedal-Schnabel schemes with various history data limits. See text what multiple-secant Byrd-Nocedal-Schnabel(1) and (2) stand for.

Scheme	History data limit					
	3	6	9	12	15	∞
multiple-secant Byrd-Nocedal-Schnabel(1)	22	19	17	17	17	17
multiple-secant Byrd-Nocedal-Schnabel(2)	22	19	17	17	17	17

Table 2 shows the iteration counts required to reach the self-consistent solution using the MSBNS(1) and MSBNS(2) schemes with the various history data limits. First, no difference in convergence is found between the MSBNS(1) and MSBNS(2) as expected. The latter, however, has an advantage. In the present study μ has not exceeded eleven. Thus in the MSBNS(2) scheme, we can economize the storage further by discarding, not ignoring, the history data formed before the $(n-\tilde{\nu}+\mu)$th iteration. Actually this feature is not implemented.

Second, as expected, the MSBNS schemes outperform the SSS and SSBNS, in which the multiple-secant condition does not hold. See also Table 1. When the SSBNS and MSBNS schemes are compared, the maximum reduction in the iteration counts is seven. This improvement, brought by the multiple-secant condition, is a less pronounced one than observed in comparison of the SSS and SSBNS schemes but a consistent one. Therefore we recommend the MSBNS(2) scheme, not only because the MSBNS(1) and MSBNS(2) schemes are no more complicated than the SSBNS, but also because in the latter the storage can be further economized.

V. Conclusion

In the present study, we have investigated the computational schemes, which are variants of the Broyden method, for acceleration of self-consistent electronic-structure calculations. In particular we have focused ourselves on the storage-saving schemes, some of which are generalized to the consideration of arbitrarily many previous iterations.

Extending the works of Srivastava and of BNS, we have proposed the two storage-saving schemes, where iteration history data are partially discarded after a prescribed storage limit is reached. As revealed in the test calculations of a Si (011) surface model, the storage-saving scheme based on BNS's formula is the superior in convergence and stability, because even when iteration history data are not kept fully, this scheme remains mathematically equivalent to Broyden's original formula.

Furthermore, starting with the SSBNS scheme, we have derived the two schemes. These satisfy the multiple-secant condition in the vector space determined on the fly so that the numerical instability is conquered.

Watching an omen of linear dependence in a given vector sequence is the key mathematical technique. Consistent reduction in the iteration counts is found. Neither is superior to the other in convergence. We have, however, pointed out that using one of the schemes may lead to further economization of the computer storage.

Finally, the BNS and related schemes have an advantage which is not exploited in the present study. When using eq. (21a) we need only a single application of $\boldsymbol{B}^{(1)}$ to a vector at each iteration step. This implies that $\boldsymbol{B}^{(1)}$ is no more restricted to a matrix which should be kept unchanged throughout the calculation. For example, we can set $\boldsymbol{B}^{(1)}$ to be a matrix representation of an electronic dielectric response function[21]. Moreover, $\boldsymbol{B}^{(1)}$ can be a more general, say nonlinear, procedure given as a set of subroutines. A self-consistent $\mathcal{O}(N)$ code[42][43], which is less accurate but runs faster, may be embedded in a place which $\boldsymbol{B}^{(1)}$ as a matrix occupied. We plan in the near future developing a code in which the traditional first-principles plane-wave pseudopotential method is combined with one of less accurate but faster methods in an above-mentioned way to achieve far more rapid convergence.

Acknowledgements

The authors wish to thank Prof. R. Yamamoto for stimulating discussions.

REFERENCES

(1) W. Kohn and L. J. Sham: Phys. Rev. **140** (1965), A1133–A1138.
(2) D. D. Koelling: Rep. Prog. Phys. **44** (1981), 139–212.
(3) G. P. Srivastava and D. Weaire: Adv. Phys., **36** (1987), 463–517.
(4) J. Ihm: Rep. Prog. Phys., **51** (1988), 105–142.
(5) W. E. Pickett: Comput. Phys. Rep., **9** (1989), 115–197.
(6) D. K. Remler and P. A. Madden: Mol. Phys., **70** (1990), 921–966.
(7) M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias and J. D. Joannopoulos: Rev. Mod. Phys., **64** (1992), 1045–1097.
(8) T. Ito: J. Appl. Phys., **77** (1995), 4845–4886.
(9) R. G. Parr and W. Yang: *Density-Functional Theory of Atoms and Molecules*, Oxford Univ. Press, Oxford, (1989), 1–324.
(10) P. Bendt and A. Zunger: Phys. Rev. B, **26** (1982), 3114–3137.
(11) R. Car and M. Parrinello: Phys. Rev. Lett., **55** (1985), 2471–2474.
(12) F. Tassone, F. Mauri and R. Car: Phys. Rev. B, **50** (1994), 10561–10573.
(13) M. C. Payne, J. D. Joannopoulos, D. C. Allan, M. P. Teter and D. H. Vanderbilt: Phys. Rev. Lett., **56** (1986), 2656–2656.
(14) A. Williams and J. Soler: Bull. Am. Phys. Soc. B, **32** (1987), 562–562.
(15) I. Štich, R. Car, M. Parrinrllo and S. Baroni: Phys. Rev. B, **39** (1989), 4997–5004.
(16) M. P. Teter, M. C. Payne and D. C. Allan: Phys. Rev. B, **40** (1989), 12255–12263.
(17) M. J. Gillan: J. Phys. C, **1** (1989), 689–711.
(18) M. Kohyama: Modelling Simul. Mater. Sci. Eng., **4** (1996), 397–408.
(19) G. Kresse and J. Furthmüller: Comput. Mater. Sci., **6** (1996), 15–50; Phys. Rev. B, **54** (1996), 11169–11186.
(20) G. P. Kerker: Phys. Rev. B, **23** (1981), 3082–3084.
(21) K.-M. Ho, J. Ihm and J. D. Joannopoulos: Phys. Rev. B, **25** (1982), 4260–4262.
(22) B. I. Dunlap: Phys. Rev. A, **25** (1982), 2847–2849.
(23) A rapidly converging scheme based on the perturbation theory is proposed also within the context of quantum device simulations. See A. Trellakis, A. T. Galick, A. Pacelli, U. Ravaioli: J. Appl. Phys., **81** (1997), 7880–7884.
(24) D. G. Anderson: J. Assoc. Comput. Mach., **12** (1965), 547–560.
(25) P. Pulay: Chem. Phys. Lett., **73** (1980), 393–398; J. Comput. Chem., **3** (1982), 556–560.
(26) C. G. Broyden: Math. Comput., **19** (1965), 577–593.
(27) G. P. Srivastava: J. Phys. A, **17** (1984), L317–L321.
(28) D. Vanderbilt and S. G. Louie: Phys. Rev. B, **30** (1984), 6118–6130.
(29) D. D. Johnson: Phys. Rev. B, **38** (1988), 12807–12813.
(30) A preliminary and daring application of the conjugate-gradient method to the traditional potential-based approach is found in X. Gonze: Phys. Rev. B, **54** (1996), 4383–4386; Convergence properties of the conjugate-gradient method are, however, not compared to those of other methods.
(31) P. H. Dederichs and R. Zeller: Phys. Rev. B, **28** (1983), 5462–5472.
(32) V. Eyert: J. Comput. Phys., **124** (1996), 271–285.
(33) A. Sawamura, M. Kohyama and T. Keishi: Comput. Mater. Sci., **14** (1999), 4–7.
(34) A. M. Rappe, K. M. Rabe, E. Kaxiras and J. D. Joannopoulos: Phys. Rev. B, **41** (1990), 1227–1230; Phys. Rev. B, **44** (1991), 13175–13176.
(35) G. Kresse and J. Hafner: J. Phys. C, **6** (1994), 8245–8257.
(36) R. H. Byrd, J. Nocedal and R. B. Schnabel: Math. Prog., **63** (1994), 129–156.
(37) J. G. P. Barnes: Comput. J., **8** (1965), 66–72.
(38) D. M. Gay and R. B. Schnabel: *Solving systems of nonlinear equations by Broyden's method with projected updates*, Nonlinear Programming 3, edited by O. L. Mangasarian, R. R. Meyer and S. M. Robinson, Academic Press, New York, (1978), pp. 245–281.
(39) R. B. Schnabel: *Quasi-Newton methods using multiple secant equations*, Technical Report CU-CS-247-83, Dept. of Computer Science, Univ. of Colorado at Boulder, (1983), pp. 1–35.
(40) G. H. Golub and C. F. van Loan: *Matrix Computation*, The John Hopkins Univ. Press, Maryland, (1993), pp. 221–233.
(41) G. H. Golub and C. F. van Loan: *Matrix Computation*, The John Hopkins Univ. Press, Maryland, (1993), pp. 128–130.
(42) T. Hoshi and T. Fujiwara: Phys. Rev. B, **52** (1995), R5459–R5462.
(43) T. Hoshi and T. Fujiwara: *Towards large-scale fully-selfconsistent LDA-electronic structure calculations (Order-N Method)*, Physics of Complex Liquids, edited by F. Yonezawa, K. Tsuji, K. Kaji, M. Doi and T. Fujiwara, World Scientific, New Jersey, (1998), pp. 129–143.